Virtual Active Network for Live Streaming Media

This application claims the benefit of U.S. Provisional Application No. 60/448,684, filed February 19, 2003.

Background of the Invention

[0001]   Delivery of streaming media and wide-area dissemination of data pose significant challenges in wide area networks such as the Internet. The large amount of bandwidth and other resources required to deliver streaming media limits the number of concurrent users. Without appropriate multicasting mechanisms, network routes may become quickly congested as a result of the same stream being delivered from its source to many recipients. The problem is compounded in wide area networks in which the load is bursty and dynamic, such as in the case of live streaming media. This can result in delays, interruptions and loss of data.

[0002]   Current approaches to solve these problems include IP level multicasting to build multicast routes from sources to sinks. This approach, however, has difficulties because of incompatibilities of various network elements of the Internet service providers and the like. As a result, some alternative approaches attempt to build overlay networks on top of the underlying physical network, and to use application level routing and multicasting through logical links between network elements, such as proxies. This approach addresses the incompatibility and interoperability problems at the physical network layer, but does not provide an appropriate mechanism for distributing load to optimize network bandwidth. Currently, each user request for data results in a data flow connection being set-up between the data origin server and the user.

However, current network infrastructures do not effectively handle congestion in the network and servers, or the changes in distribution of end user populations. Therefore, media or data streams can suffer from network congestion on delivery paths. Accordingly, an approach to this problem has been to use proxy caching for media data delivery. This approach treats media data as an object to cache at edge caches for delivery to nearby end users. It is useful for video clips and the like, but is not suitable for live broadcasting of streaming media. Other approaches use pre-configured proxy networks. However, these do not efficiently accommodate changes in system load or user distributions, and do not efficiently handle live streaming media which has bursty traffic conditions at the beginning of an event.

[0003]   There is a need for systems and methods that address these and other problems of efficiently distributing live streaming media and other data in wide area networks, and it is to these ends that the present invention is directed.

Summary of the Invention

[0004]   The invention affords an application level proxy network architecture and method for distribution of live streaming data and wide area data dissemination that aggregates routes between data sources and sinks. The invention provides a hierarchical overlay network structure that may be automatically and dynamically adjusted based upon conditions such as user population distribution, usage patterns, and network conditions. The system architecture affords reliable and high quality live streaming media delivery, lower server resource requirements at the content provider sites, reduced inter-ISP traffic, application level routing for rapid deployment and cost-effective media data delivery.

[0005] In one aspect the invention affords a method of distributing streaming data in a wide area network that has an overlay network of proxy servers that comprises activating the proxy servers to form a hierarchical structure comprising multiple tiers of proxy servers with respect to a data stream from a corresponding data source to distribute the data stream to a plurality of users. The proxy servers are activated in the multiple tiers based upon the users and in order to provide predetermined network operating conditions. The hierarchical structure is dynamically reconfigured as users change in order to maintain the predetermined network operating condition.

[0006] In another aspect, the invention distributes streaming media in a wide area network by activating proxy servers of an overlaid network to form first and second hierarchical structures in multiple tiers to distribute corresponding first and second data streams to first and second groups of users, respectively. The first and second hierarchical structures share one or more proxy servers of the overlaid network of proxy servers, and the numbers of tiers and proxy servers in each tier of the first and second hierarchical structures is based upon the first and second groups of users, respectively. The hierarchical structures are then reconfigured as the groups of users change. The first and second hierarchical structures may share one or more proxy servers of the overlaid network of proxy servers.

[0007] In a further aspect, the invention provides a method of distributing streaming data in a wide area network having an overlay network of proxy servers that comprises activating the proxy servers to form a hierarchical structure comprising multiple tiers of proxy servers in order to provide a data stream from a corresponding data source to a plurality of users. The proxy servers are activated by predicting a rate of logon of users to the network, and activating a group of proxy servers in one tier as a server farm. Users logging on to the network are distributed to

the proxy servers of the proxy farm in a manner so as to balance the data loads of the proxy servers. The hierarchical structure is dynamically reconfigured as users change in order to maintain a predetermined operating condition of the network.

[0008]   The invention automatically and dynamically adjusts the collaborative proxy network hierarchical structure to account for varying conditions without the need for human operators. This dynamic adjustment may be based on parameters that include end-user population, geographical distribution of user requests, network conditions, and location and capacity of proxy servers, and varying loads.   As demand (load) increases, additional proxy servers may be added to the active network and the data connections redistributed. Similarly, the network proxies may use a peering arrangement with other proxies to consolidate live connections when the workload shrinks. A proxy network coordinator (PNC), a logical entity that can be implemented centrally as a single component or in a distributed fashion across multiple components, is used to determine appropriate routes across the proxy network for delivering data streams.   In contrast to known approaches that are architected in the network/service layers, the virtual active network of the invention is architected at the application layer.   Application level protocols among network proxies are used to support efficient distribution of live data streams.

[0009]   In addition to ease of deployment, since the routing scheme of the invention is based on application level functions rather than network level functions, a significant advantage of the invention is that it is capable of handling live media broadcasts. It is especially adaptable to deal with the bursty characteristics of multiple user logins (and user logoffs). Furthermore, unlike most other approaches which assume that proxy activation is instantaneous upon request, the invention specifically accounts for the delay involved in proxy activation and connection migration, thereby ensuring no loss of data.

Brief Description of the Drawings

[0010]   Figure 1, comprising Figures 1(a) - (b), illustrates respectively, diagrammatic views showing the architecture of a proxy network in accordance with the invention deployed in a wide area network such as the Internet, and the proxy network arranged as a three-tiered overlaid network;

[0011]   Figure 2, comprising Figures 2(a) - (d), illustrates a load distribution process in accordance with the invention for distributing expanding loads to proxy servers arranged in a three-tiered hierarchical structure;

[0012]   Figure 3, comprising Figures 3(a) - (c), illustrates overlays of the same set of cooperating proxy servers to serve multiple sources of data;

[0013]   Figure 4, comprising Figures 4(a) - (c), illustrate a load consolidation process in accordance with the invention to handle reducing loads;

[0014]   Figure 5 illustrates the dynamic allocation of proxy servers for load distribution in a bursty environment;

[0015]   Figure 6 illustrates a portion of the internal architecture of a proxy server;

[0016]   Figure 7 illustrates a process for initializing a virtual active network;

[0017]   Figure 8 illustrates a proxy server process for handling media streams;

[0018]   Figure 9 illustrates a process for handling a login event;

[0019]   Figure 10 illustrates a process for a logoff event;

[0020] Figure 11 illustrates a module comprising data structures by which a proxy network coordinator maintains information on the dynamic relationships among proxy servers;

[0021] Figure 12 illustrates a DISTRIBUTE process by which a proxy network coordinator distributes loads among proxy servers;

[0022] Figure 13 illustrates a process by which a proxy network coordinator creates a proxy server farm; and

[0023] Figure 14 illustrates a CONSOLIDATE process by which a proxy network coordinator consolidates proxy servers in a decreasing load environment.

Description of Preferred Embodiments

[0024] Figure 1(a) illustrates the architecture of a proxy network in accordance with the invention comprising a plurality of proxy servers P_{11}- P_{33} deployed in a wide area network 20 such as the Internet. As shown, the proxy network may also include a proxy network coordinator (PNC) 24, and a plurality of network routers 26. During an initialization phase when a media server S is introduced to the network, the proxy network may be partitioned into a hierarchical virtual active network (VAN) structure comprising multiple tiers of proxy servers based on conditions such as the population and distribution of end users $u_1 - u_6$ and the relative distances among the media server, proxy servers and end users, and data loads. This is preferably done by and under the control of the proxy network coordinator (PNC) 24 which coordinates the connections between the proxy servers, as will be described.

[0025] Figure 1(b) shows the proxy network of Figure 1(a), in which proxy servers P_{11}-P_{33} are arranged in a three-tiered hierarchical network structure which comprises a single data

source, server (S) 28, and proxies P_{11}-P_{13} arranged in a Tier 1, 31; proxies P_{21}-P_{23} arranged in a Tier 2, 32; and proxies P_{31}-P_{33} arranged in a Tier 3, 33. P_{ij} indicates the jth proxy server in the ith tier of the overlay network. Proxy servers in a higher tier (lower tier number) of the hierarchical network structure are referred to as "parents", and servers or users in a lower tier of the hierarchical network structure are referred to as "children".

[0026] The links 30 between components shown in the overlay network are all logical; the actual communication between two proxy servers still requires routing at the level of the network routers 26. End users, u_i, may connect to the overlay network proxies via domain name (DNS) resolution based redirection.

[0027] In this specification, the term "overlay" refers to a network of proxy servers ("proxies") deployed strategically on top of an existing network as shown in Figure 1(a); an "overlay network" refers to the static partitions of the proxies organized in a multi-tier hierarchy structure with respect to a given data stream as shown in Figure 1(b); and a "virtual active network" refers to the live or active components of an overlay network connected by links such as links 30.

[0028] Although Figure 1 illustrates an overlay network with a single media server S, as with typical network routers, each proxy server 24 can also serve multiple media streams originating from one or more media sources. Also, a single physical proxy server can be shared by multiple VANs, each for a different media source, or for multiple streams from the same source. Figure 3 shows an example of overlay network architecture consisting of nine proxy servers shared by two media servers, S1 and S2, to deliver streaming data to two different groups of users, i.e., u_1-u_7 and u_8-u_{14}. The solid lines denote the streams from the server S1 and indicate

a first VAN, and the dashed lines denote the streams from the server S2 and indicate a second VAN. As shown, many virtual proxies, for example, P_{11} of the first VAN for data streams from S1 (Figure 3(b)) and Q_{3l} of the second VAN for data streams from S2 (Figure 3(c)) share the same physical servers. The virtual active network for each data stream may also have a different number of tiers. As shown in Figures 3(b) and 3(c), the numbers of tiers of the virtual active networks for S1 and S2 are three and four, respectively, and the number of proxy servers in each tier may be different.

[0029] In the VAN architecture of the invention, redundant retrieval capability is utilized during restructuring of the multicast network. When a proxy needs to change its parent due to varying network conditions, the proxy establishes a connection with the new parent proxy before disconnecting from the old parent proxy. Since the traffic between two proxy servers is more crucial than the traffic between a proxy server and end users (loss of a single inter-proxy connection may affect multiple users adversely), a proxy server may retrieve multiple streams of the same data from the proxy servers in its parent tier. This ensures a higher quality of streaming data delivery.

[0030] A proxy server that is serving a stream is an active proxy server (with respect to that particular stream). A proxy server that is active with respect to a given stream may operate in different phases, i.e., an expansion phase, a contraction phase, or an idle phase. When a proxy server is activated by the PNC, it is in the expansion phase until the PNC initiates a contraction phase. During the expansion phase the proxy server continues to accept new connection requests until its load reaches a predetermined threshold, e.g., three data sinks. In response to an active proxy server notifying the PNC that its load has reached a given threshold level, the PNC will perform a consolidation process to redistribute the load, and will activate additional proxy

servers in the same tier or a higher tier to serve new or migrated traffic. After such a load distribution operation, a proxy server transitions from the expansion phase to the contraction phase, and will cease receiving new connection requests. Subsequently, when the load of a proxy server drops below a given threshold, the proxy server requests consolidation from the PNC. When the load falls to zero with respect to a given data stream, the server becomes idle.

[0031]   The proxy network coordinator (PNC) is a logical entity that can be implemented centrally as a single computer 24 (as shown in the figures) or in a distributed fashion across multiple network components. The PNC coordinates the connections between proxy servers using load distribution and load consolidation processes as will be described shortly. For each streaming data server S1, S2, the information the PNC maintains in order to establish and dynamically manage the VANs may include, for example, the numbers of tiers and proxy servers in each tier; the network status between proxy server pairs in adjacent tiers; a list of active proxy servers in each tier; and the hierarchical structure of the virtual active network as identified by proxy server pairs in adjacent tiers.

[0032]   A principal task of the PNC of the invention is to maintain the hierarchy structure of the VAN for each media server. It does this by dynamically allocating and reallocating resources in response to messages from the proxy servers to adapt to changing network conditions, loads and events. During an initialization phase, the static overlay network of proxy servers may be initialized by a PNC associated with a media source into a VAN by activating one proxy server at each tier in preparation to forming a connection path across the overlay network for the media stream. The PNC may also activate multiple proxy servers in response to actual or anticipated network conditions. The actual establishment of parent-child relationships among the proxies occurs during dynamic restructuring of the virtual active network by DISTRIBUTE and

CONSOLIDATE processes, as will be described. An active proxy server initiates a DISTRIBUTE process by sending a DISTRIBUTE message to the PNC when its load reaches a selectable maximum threshold. In response the PNC activates one or more proxies in the same tier. Similarly, a proxy server initiates a CONSOLIDATE process by sending a CONSOLIDATE message to the PNC when its load falls below a selectable minimum threshold. This message indicates to the PNC that the proxy should be made idle or dormant. A sequence of DISTRIBUTE and CONSOLIDATE processes causes the proxy hierarchy structure of the VAN to expand and contract dynamically to meet changing conditions. These PNC processes will be described in detail in connection with Figures 12-14.

[0033] The PNC may activate the minimum number of proxies required at each tier to ensure coverage of all anticipated endusers for a given media event. The IP addresses of proxies to which the endusers should be redirected when they request for the media stream is registered using the well-known Domain Name Service, DNS, system. The IP address of each end user is maintained in the proxy servers while the proxy network hierarchical information is maintained only at the PNC. When the number of enduser logon requests to a proxy server increases to the predetermined maximum connection threshold, the proxy server may send a DISTRIBUTE request message to the PNC to expand the VAN hierarchy by adding additional proxies. The number of proxies activated in response is preferably based on the rate at which endusers arrive onto the network (which can be determined as will be described below). Similarly, when endusers logoff and the connections to a proxy server decrease to a minimum connection threshold, the server may send a CONSOLIDATE request message to the PNC. In response, the PNC contracts the VAN hierarchy to redistribute connections and minimize bandwidth usage. Due to the hierarchical structure of the VAN, most of the changes tend to occur in the lower tiers

of the overlay network. The number of changes decreases significantly in the upper tiers in the overlay. This advantageously results in a media server being cushioned from the adverse effects of abrupt changes in the network structure and loading. Figures 2 and 4 illustrate the manner in which the VAN structure expands and contracts.

[0034] Figures 2(a) through (d) show an example of a load distribution process for expanding a VAN. In this example shown, there is a single source server (S) 28 for the media source data. Below the source server there are the three Tier 1 proxy servers P_{11}, P_{12}, and P_{13}. Below the Tier 1 proxy servers in the structural hierarchy, there are Tier 2 proxy servers P_{21}, P_{22}, and P_{23}. At the lowest Tier 3 there are proxy servers, P_{31}, P_{32}, and P_{33}. Below Tier 3 are the end users u_1-u_7. For purposes of the following explanation, the load capability of each proxy server may be assumed to be limited to three simultaneous connections.

[0035] At a first time, represented by Figure 2(a), when user u_1 wishes access, it is directed by the DNS mechanism to send a request to P_{31}. The PNC causes one proxy (P_{11}, P_{21}, and P_{31}) at each of Tiers 1-3 to be activated to form a streaming path 40 between the media source server 28 and user u_1. As users u_2 and u_3 request access, streaming paths 41 and 42 are provided by P_{31}, as shown in Figure 2(b). When user u_3 arrives, however, P_{31} reaches a maximum threshold corresponding to the limit of its assumed capacity (in this example) and it sends a DISTRIBUTE request to the PNC (not shown in Figure 2). In response, the PNC may select P_{32} from the overlay network and activate it by sending a message to P_{32} to indicate that it has been activated as part of the VAN and that its parent server is P_{21}. PNC then updates the DNS resolution mechanisms so that later users u_4-u_6 are directed to P_{32} instead of P_{31} (Figure 2(c)). The arrival of u_6 brings the connections to P_{32} to its threshold of three (assumed in the example), and will trigger a DISTRIBUTE request by P_{32} to activate a new proxy. The arrival of u_7 (Figure 2(d))

will similarly trigger a DISTRIBUTE request by P_{2j}, which now is at its assumed capacity, to activate a new proxy in Tier 2. This sequence of events illustrates the process by which the virtual active network of the invention expands gracefully as the network load increases.

[0036]   Figures 4 (a) - (c) show an example of load redistribution in a contracting VAN. As the number of users in the network drops due to log offs, the VAN structure has to contract by deactivating proxy servers so that the proxies and the network resources are not underutilized and the network bandwidth is optimized. This load redistribution process is referred to as CONSOLIDATION, and is also controlled by the PNC.

[0037]   Figure 4(a) shows a hypothetical configuration of the VAN (corresponding to that shown in Figure 2(d)) with active proxies P_{11}, P_{21}, P_{31}-P_{33} and users u_1-u_7. When users u_4, u_2, and u_3 log off one after another (Figure 4(b)), the reduction in the load at P_{31} triggers a CONSOLIDATE request from P_{31} to the PNC (not shown). In response, the PNC executes a CONSOLIDATE process (as will be described below) by sending a message to the children of P_{31} (u_1 in this case) to switch to another proxy server. Preferably, the switch is to the most recently activated proxy server, i.e., P_{33}. Consequently, u_1 logs off from P_{31} and logs on to P_{33}, which results in P_{31} logging off from P_{21} (Figure 4(c)).

[0038]   Allocating proxy servers one at a time to expand a VAN as described in connection with Figure 2 may not be acceptable when traffic is bursty and rapidly changing, such as, for example, at the beginning of a media event when most of the users login to the network. To deal with the need for fast restructuring, the PNC may tune the allocation rate of new proxy servers to deal with anticipated loads and bursty traffic. This may be done by estimating the rate of arrival of users, the capacity of the servers to handle loads, and the rate at which new servers will be

required to be activated to provide the needed capacity to handle the anticipated load. An preferred example of a tuning process is illustrated in Figure 5, and will now be described.

[0039] When the PNC receives a DISTRIBUTE message from a proxy server, the PNC may compute the rate of arrival of new users as:

$$New\backslash\_User\backslash\_Arrival\backslash\_Rate = \frac{Number\_of\_\Pr oxies\_assigned}{current\_time - Last\_DISTRIBUTE\_request\_time}.$$

[0040] PNC next computes the new *average* user arrival rate as:

(New_User_Arrival_Rate + α × Average_User_Arrival_Rate).

[0041] The value of the parameter α may be selected to provide a desired tuning, and its value may be fixed or dynamically changed according to network conditions. A value of 1 for α treats all access patterns equally, while a value of 0 considers only the current user arrival pattern. The PNC then computes the number of proxy servers needed to be activated for the new user arrival rate as follows:

$$Number\backslash\_of\backslash\_\Pr oxies\backslash\_assigned =$$
$$Round\left(Number\backslash\_of\backslash\_\Pr oxies\backslash\_assigned \times \frac{Average\_User\_Arrival\_Rate}{Old\_Average\_User\_Arrival\_Rate} \right).$$

[0042] When more than one proxy server is activated at the same time, the group of proxy servers functions like a server farm. The PNC may then distribute the connection requests from end users and proxy servers to the group of activated proxy servers in the server farm in a way to balance their loads, such as in a "round-robin" fashion. As soon as any one of the proxy servers in this server farm group sends a DISTRIBUTE request to the PNC, the PNC treats this request

as a collective distribute request from all of the servers. The rationale for this is that if the requests are distributed to all activated proxies in a round-robin fashion, then all proxy servers will be equally loaded. The arrival rate formulation may therefore be adjusted to handle simultaneous arrivals of multiple DISTRIBUTE requests. Furthermore, the PNC may deactivate all the proxies that were activated at the same time to minimize the generation of redundant DISTRIBUTE events.

[0043]   The server farm approach is in contrast to distribution to a single server. When a single proxy server is serving live streams from multiple media servers, the DISTRIBUTE events are preferably treated independently. That is, a DISTRIBUTE request by a proxy server on behalf of a media source does not impact the state of other media sources at that proxy server. In order to achieve this independence, the load control parameters may be dynamically adjusted when a media source handling is included or by the proxy server.

[0044]   Figure 6 illustrates a portion of the relevant logical architecture of a typical proxy server 100, such as P23. The proxy server may physically comprise a computer. A main resource of the proxy server is a buffer memory 110 used for storing the streaming media. The buffer is preferably shared and accessed by an incoming stream handling module (ISHM) 120 and an outgoing stream handling module (OSHM) 130. ISHM 120 interfaces to a media server or to parent proxy servers in the tier immediately above the proxy server 100 in the overlay network, and it receives media streams from the media server or from parent proxy servers. ISHM is responsible for managing connections, disconnections and reconnections to the parent proxy servers, as specified by the PNC. Preferably, it has the capability of connecting to multiple parents for access to multiple sources of a given data stream to enable redundant and, therefore, robust, media delivery. As shown in the example of Figure 6, proxy server 100 (P23)

may connect to three parent proxies P11, P12 and P14. ISHM may fetch media streams from P11, P12 and P14 block-by-block, and store the blocks in the buffer memory 110 as, for example, in block order as shown. ISHM may eliminate redundant blocks received from its parent proxies, such as one of Blocks 23 from P11, P12 and P14, and one of Blocks 22 from P12 and P14, by checking either or both of the block sequence numbers or time stamps. After eliminating redundant blocks, a retained block is stored in the buffer memory 110.

[0045]   OSHM 130 interfaces to the buffer memory as well as to child data sinks, such as users u_1-u_5, and proxies P31-P33, in the tier immediately below proxy server 100. The OSHM provides streaming media data to the users and down-stream child proxies.

[0046]   Proxy server 100 also keeps track of the end users and child proxy servers in the next lower tier that are retrieving the streaming media through connections to the proxy server 100. This may be done by tracking the IP addresses of the end users and down-stream proxy servers who request data streams from proxy server 100. These IP addresses may be extracted from the media protocol, such as the RTSP, headers. When redirection of end users and proxy servers in the lower tier is needed, proxy server 100 may send out redirection messages to its connected end users to switch to a newly assigned proxy server, while the proxy server coordinator (PNC) sends messages to redirect the child proxy servers in the lower tier to the new parent proxy server. Accordingly, proxy server 100 may maintain the IP address of each connected end user, while the PNC maintains proxy network hierarchical structure information. Moreover, since the VAN architecture permits redundant retrieval of streaming media during the restructuring of a multicast network, when proxy server 100 needs to change its connection to a new parent due to changing network conditions, it first establishes a connection with the new parent before disconnecting from the old parent proxy server. Additionally, as described above in connection

with Figure 3(a), one proxy server may be shared by multiple VANs for handling different media sources, or for handling multiple streams from the same source. Therefore, the proxy server also maintains other information related to its sink and source connections and the network, as indicated in Figure 6.

[0047] Referring to Figure 6, Stream ID designates a unique identification, e.g., a URL, for each media stream handled by the proxy server. As shown in the example, proxy server 100 may handle media streams from two different sources at URLs "www.ccrl.com" and "www.nec.com". Proxy server 100 may also handle three different media streams, i.e., "demo1.rem", "demo2.rem" and "demo3.rem" from source www.ccrl.com. For delivering the three different media streams, only a single virtual active network tiering assignment is necessary. All proxy servers in a particular VAN will have the same prefix, e.g., P, for source www.ccrl.com. On the other hand, a separate virtual active network tiering assignment, and proxy prefix such as Q, will be required for handling media from source www.nec.com. Furthermore, although media streams demo1.rem, demo2.rem and demo3.rem may be delivered through the same VAN, the proxy server assignment in each tier may be different. If the geographical distribution of end users for the three media streams is a wide one, it may be beneficial to employ different virtual active network tiering assignments for the servers. While proxy servers may be logically partitioned into multiple virtual hosts, as shown in Figure 6, the number of simultaneous connections and the bandwidth constraints of the server needs to be enforced on a machine-wide basis.

[0048] As shown in Figure 6, proxy server 100 may also maintain information related to the hierarchical structure of the proxy network such as the IP addresses of upstream and down-stream proxy servers, users and the PNCs. Figure 6 shows, for example, that proxy server 100

maintains information on "Disconnecting parents: P11" (P11 is disconnecting as indicated by the dotted line between P11 and the ISHM); "Connecting parents: P12"; "Connected parents: P14"; the IP addresses of the end users which are logged in at the proxy server, as well as children and forwarding proxy servers. Other information may include the physical capacity corresponding to the number of downstreams that the server can support and its logical capacity corresponding to the maximum number of downstreams assigned to the proxy server.

[0049]   As previously noted, the PNC determines the hierarchical structure of a VAN by the allocation of individual proxy servers into tiers within that structure. It is desirable that this structure and the allocation of proxy servers be such that the utilization of data network resources, e.g., bandwidth, be optimized for efficiency and data integrity. A preferred approach to accomplishing this is to partition the data network into geographical regions and to assign proxy servers and the tiering structure in each region to users located in that region, for example, by directing ISPs to servers in their region, upon initialization of a VAN. This may be accomplished by determining the proxy servers in each region, and the layers (tiers) to service the users in the region. This approach is illustrated in Figure 7.

[0050]   As shown in Figure 7, proxy servers 201-204 in a first region (Region1) may be activated in a four-tier hierarchical structure to provide media data from a media server S to users $u_1 - u_4$ located in region R1. Similarly, proxy servers 301-303 may be allocated to a second region R2 to serve users u_5-u_8 located in that region, and proxy servers 401-404 may be allocated to a region R3 to serve users u_9-u_{11} in region R3. Using the media server S as the media source and the center of a fanout, the overlay structure of the VAN for that particular media source may be determined based upon the number of the autonomous data systems or ISPs in each region, the connectivity information among the data servers and the size of each for each region. This

task may be performed during run time and adjusted by region and layer partitions based upon network conditions. The PNC may then affect the overlay network structure by sending the appropriate URLs to the various proxy servers to identify parent and child proxy servers.

[0051]   Figure 8 illustrates a proxy server module referred to as "DynamicMultiSourceProxyServer" that provides data structures for maintaining connection states of multiple media streams and message-based APIs for communicating with other proxies and the PNC. As shown, the module may comprise two data structures for maintaining information on the connection states of the multiple data streams and on the current parent proxies to which a server must connect for each stream source.

[0052]   The first data structure "ConnectionState" shown in Figure 8 maintains the current state of all live connections that are passing through a given proxy. This includes the URL of the stream source for which the connection is maintained, the IP address and other connection related information of the parent to which the proxy is connected, and the IP addresses of all the child hosts (either another proxy or an end-user) that are being served by the proxy server. The PNC may offload the task of end-user maintenance to the proxy server itself. The second data structure "ProxyParent" shown in Figure 8 maintains information on the current parent proxy to which the proxy server must connect for each stream source.

[0053]   Two main events that a proxy server must support are LOGIN and LOGOFF requests from users. Depending upon the active network structure, a user may be another proxy server, e.g., a child proxy, or an enduser. Other processes shown may be used to maintain the virtual active network for each stream source. A process "SwitchToParent" may be triggered by the PNC if the PNC detects that there are network problems or if the proxy load needs to be

balanced. Similarly, another process, MONITOR, may be initiated by the PNC so that the proxy can monitor network links between consecutive tiers in a distributed manner. The PNC may require, for example, that for each stream source a proxy server in tier k+1 monitor all the parent proxies in tier k.

[0054] Figures 9 and 10 illustrate, respectively, the LOGIN and LOGOFF processes which are performed by proxy servers in response to login and logoff events. The LOGIN process (Figure 9) may be triggered either by another proxy or by an end-user. If the event is triggered by a proxy, this means that the proxy has been assigned to be a parent proxy for the specified media stream. As shown in Figures 8 and 9, upon receiving a LOGIN request from a sender S for a sourceURL, a check is made to determine if the connection is already active for the specified sourceURL. If not, the connection is initiated by setting up a local data structure and uploading the sender's request for a connection to the parent proxy. Also, if a new media begins to share the proxy server, the server dynamically adjusts the load control parameters to account for the fact that there is more contention for physical resources at the proxy. If the connection is already active then the LOGIN request is served locally by including the request sender S in the local connection state. If the server's load exceeds a predetermined maximum threshold, the server sends a DISTRIBUTE request to the PNC for additional proxies to be activated in its tier. The load condition for the DISTRIBUTE event may be determined by the login rate and the logoff rate. If the two rates are such that in a next predetermined time period, ΔT seconds, the proxy may reach its maximum capacity, the proxy triggers a DISTRIBUTE request to the proxy network coordinator. The parameter ΔT may be set to afford sufficient time for the PNC to activate new proxy servers to minimize loss of data.

[0055] The LOGOFF process shown in Figure 10 is analogous to the LOGIN process just described. When a sender S triggers this event at a proxy, the proxy removes S from the connection state and updates the data structures of Figure 8. If S is the last child to request logoff, the proxy sends a LOGOFF message to its parent server since it no longer needs a connection. If the workload falls below a certain threshold due to multiple logoff events, the proxy server notifies the PNC that it should be made dormant and move to an idle state by a sending a CONSOLIDATE message. On receiving the message, the PNC deactivates the proxy by moving its connections to another active proxy server in the same tier. If that proxy server is the only remaining proxy server serving the content stream in the same tier, the PNC will ignore the CONSOLIDATE message.

[0056] Figures 11-14 illustrate the data structures and processes which may be employed in the proxy network coordinator for maintaining the application level proxy network for multiple media streams from different sources. Figure 11 illustrates a module designated as "DynamicMultiSourcePNC", which includes data structures for maintaining static network information and the dynamic relationships among the proxies. A message-based API may be used by the proxy servers to coordinate their activities. A data structure SourceProxyPair maintains information on the relationship between stream sources and the corresponding proxy servers.

[0057] The DISTRIBUTE process is illustrated in Figure 12 and the CONSOLIDATE process is illustrated in Figure 14, and these will be described shortly. First, however, more description of the dynamic adaptive resource allocation afforded by the invention will be described.

[0058]   In networks of the type with which the invention may be employed, user traffic may

ordinarily be directed to an arbitrary set of servers. There is an associated time delay in

redirecting the traffic between the servers due to the time required to set routers and load the

media. This time delay for the redirection process may be designated as ΔT. Additionally, in

order to afford network stability, it is undesirable to remove a server from the network if it will

be needed a short time later. A stability parameter "sp" may be used as described in more detail

below to control stability of the network. When a server is added or removed, the invention

attempts to ensure that during the next period sp there will not be a need to change the number of

servers in the network. Also, when a server expects that during the next time ΔT its load will

exceed its capacity, it may send a DISTRIBUTE request to the PNC. The parameter ΔT is

preferably selected to provide sufficient advance warning to the PNC so that it may redirect users

to an available server before some users are rejected due to an overload.

[0059]   Figure 12 illustrates the DISTRIBUTE process. When the PNC receives a

DISTRIBUTE request from a proxy, it first checks whether during the next *sp* time the capacity

available in the network will be sufficient to serve the users that request the particular media

stream. If not, it adds the minimum number, m, of servers that will be necessary to handle the

load during the *sp* period. If the PNC is unable to activate the anticipated number server needed,

it may activate as many servers as it can, and group them into a server farm as previously

described.

[0060]   If the PNC expects that the overall server capacity will be sufficient for the next sp

period, the PNC may try to find a suitable proxy or group of proxies to which it can redirect

traffic to optimize the network. By re-grouping the existing servers, the PNC may overcome the

restrictions due to fragmentation by re-grouping the proxies so each proxy will have sufficient

capacity to handle new users during the next ΔT period. It may do this by estimating whether servers have the capacity to handle the expected loads in the following manner.

[0061] In Figure 12, "SysLinRate" denotes the average predicted login rate and "Loffrate$_s$" denotes the average predicted logoff rate for each proxy server S. Each server in a group of m servers will observe SysLinRate/m logins. If a proxy can handle the load assigned to it, then either the logoffs are higher than logins ((SysLinRate/m) < Loffrate$_s$) or it has available space for ΔT period ($\Delta t * $ ((SysLinRate/m) – Loffrate$_s$) < Max$_s$ – Load$_s$). Therefore, for any given proxy S, the minimum m (MIN$_s$) for this condition can be easily calculated. After calculating these MIN$_s$ values, each proxy can be hashed according to those values. An eligible group of size l exists if the number of the proxies that has MIN$_s$ value less than or equal to l is greater than or equal to l. Since any suitable minimum size group is acceptable, the first l servers may be chosen. Min l and MIN_s values can be evaluated for the worst case based upon the number of active proxies. The procedure used to create a server farm group is shown in Figure 13.

[0062] Figure 14 illustrates the PNC CONSOLIDATE process to redistribute the load and idle proxies when the network load decreases. When a proxy expects to reach to zero load level in a short time, as determined by the login and logoff rates, the stability parameter sp, and the loads, it may send a CONSOLIDATE request to the PNC. The sp parameter may be adjusted to achieve a desired rate of consolidation. In any case, a proxy must be in a contracting mode, i.e. log-offs must be higher than logins, to send a CONSOLIDATE request.

[0063] When the PNC receives the CONSOLIDATE request, it checks whether the consolidation of this server will necessitate the creation of a new server in the next *sp* time period. It also checks whether there is enough space in the currently active proxies to handle

additional traffic after consolidation. If both of the conditions are satisfied then the proxy is consolidated. Otherwise, the request is ignored.

[0064]   As mentioned above, the adaptive resource allocation methods of the invention depend on predicting the expected number of users in a predetermined time period. Preferably, the time period is selected to afford good predictability. The invention does not try to predict very far in advance, and a simple time-series prediction method such as double exponential smoothing has been found effective in predicting login and logoff rates. Double exponential smoothing is used also because it is easy to implement and it is very efficient to execute. Double exponential smoothing use the following two equations to predict the future values.

$$S_t = \alpha y_t + (1 - \alpha)(S_{t-1} + B_{t-1}) \tag{1}$$

$$b_t = \beta(S_t - S_{t-1}) + (1 - \beta)b_{t-1} \tag{2}$$

where y_t is the last value observed and $0 < \alpha, \beta < 1$. Suitable values are $\alpha = 0.9$ and $\beta = 0.1$. S_t is the value of the average future prediction.

[0065]   The parameters of the double exponential smoothing are preferably updated every $\Delta t/10$ period, based on the number of logins in the last $\Delta t/10$ period. For example, if Δt is 10000 milliseconds (ms) and 300 users logged in in the last second, the last login value observed will be (y_t) (300/1000) per ms. The new S_t value calculated will be used as the prediction of average number of users login per ms. To predict the values for at least Δt time in the future, it may be desirable to update the predictions more often then Δt to capture any sudden changes in user pattern. However, it should not be updated so frequently as to create a bottleneck. The choice of $\Delta t/10$ period results from these considerations.

[0066]   As discussed earlier, a proxy in a child tier monitors the network connectivity to all

the proxies in the parent tier based on the static overlay structure. If the network connectivity

from child proxy S to a parent proxy P is lost, S sends a LinkDowngrade event to the PNC. In

this case, PNC first tries to find an alternate parent proxy P' for S' and sends a message

SwitchToParent to S asking it to establish P' as the new parent. If an active parent other than P

does not exist in that case a sibling proxy S' is located and the children of S are migrated to S'. If

both P' and S' are non-existent then in that case PNC activates a proxy P" in the parent tier

(similar to DISTRIBUTE) and asks S to switch to P". In addition to monitoring the network

links, the PNC may monitor the network host status of each proxy in the system. If the PNC

detects that a proxy server has crashed (through a timeout mechanism, not shown), then the PNC

may migrate all the children of failed proxy with respect to each media type to an alternate active

proxy in the same tier. If no active proxy exists, then a proxy is activated in the specified tier.

Through the above message-based events, the PNC dynamically maintains the structure of the

virtual active network.

[0067]   The foregoing has described preferred embodiments for a peer-to-peer virtual active

network (VAN) network architecture for streaming data delivery in which a plurality of proxy

servers in a hierarchical structure are coordinated to deliver media streams. The hierarchical

structure is dynamically reconfigured based on network conditions, such as to optimize (e.g.,

minimize) bandwidth and improve network performance. It will be appreciated that changes in

these embodiments may be made without departing from the spirit and principles of the

invention, the scope of which is defined in the claims.